

Промежуточный отчет по программному проекту

1. Основные планы и этапы проекта

1.1 Название проекта: Конструктор смарт-контрактов с динамической комиссией для децентрализованных бирж

Краткое описание проекта: Проект представляет собой библиотеку хуков (смарт-контрактов) для протокола Uniswap V4, реализующую алгоритмы динамического изменения комиссий в пулах ликвидности. Решение автоматически корректирует размер комиссии в зависимости от рыночных условий (волатильности, объема торгов, арбитражного спреда), что снижает непостоянные потери для поставщиков ликвидности и повышает эффективность DEX. Включает инфраструктуру для тестирования, развертывания и веб-интерфейс для конфигурации.

Цель проекта: Снижение непостоянных потерь поставщиков ликвидности через внедрение и тестирование алгоритмов динамических комиссий в децентрализованных биржах на базе Uniswap V4.

Краткое описание (декомпозиция) задач:

- Исследование и проектирование:
 - Изучить математические модели АММ, принципы работы Uniswap V4 и хуковой системы.
 - Проанализировать существующие подходы к динамическим комиссиям и непостоянным потерям
 - Определить архитектуру библиотеки и интерфейсы для унификации хуков.
- Разработка ядра (смарт-контракты)
 - Реализовать базовое наследование от контракта BaseOverrideFee с унифицированным интерфейсом для различных видов динамических комиссий
 - Реализовать несколько конкретных типов хуков (например, на основе истории прошлых свопов, объема торгов, текущих цен на централизованных биржах СЕX)

- Интегрировать хуки с оракулом Chainlink для получения внешних ценовых данных
- Обеспечить совместимость с артефактами Uniswap V4 (SwapRouter, PoolManager, PositionManager)
- Создание инфраструктуры
 - Реализовать скрипты (Foundry) для автоматического развертывания всей инфраструктуры (артефакты Uniswap V4, хуки, пулы) на локальном блокчейне (Anvil)
 - Написать комплексные тесты, симулирующие торговую активность и изменение комиссий
- Разработка веб-интерфейса и инструментов
 - Создать веб-интерфейс для выбора типа комиссии, настройки параметров и генерации кода контракта
 - Реализовать функционал сборки и скачивания готового проекта для развертывания
 - (Опционально) реализовать сравнение эффективности разных типов комиссий на основании исторических данных с Binance

1.2 Планы и этапы выполнения проекта:

Этап проекта	Описание работ	Ожидаемые результаты	Сроки выполнения
Постановка задач и определение требований	Формулировка задач и их декомпозиция	Утвержденная декомпозиция задач	До 30.11.2025
Техническое задание 1	Подготовка документации	Оформленное техническое задание	До 16.12.2025

Разработка ядра библиотеки хуков	Реализовать базовое наследование от контракта BaseOverrideFee и 1-2 первых алгоритма динамических комиссий на Solidity. Интеграция с хуковой системой Uniswap V4 и базовое тестирование на локальном блокчейне (Anvil)	Функционирующие смарт-контракты хуков, способные корректировать комиссию пула в соответствии с заданной логикой, и проходящие базовые тесты на взаимодействие с Uniswap V4	Декабрь 2025
Разработка инфраструктуры тестирования и развертывания	Создание скриптов (Foundry) для автоматического развертывания всей экосистемы (артефакты Uniswap V4, хуки, пулы ликвидности). Разработка тестов, симулирующих торговую активность и изменение комиссий. Интеграция с Chainlink Price Feeds.	Готовая к использованию инфраструктура для локального тестирования, автоматического развертывания смарт-контрактов	Январь 2026
Базовый web-интерфейс	Разработка веб-интерфейса для конфигурации параметров хука, генерации Solidity-кода и сборки готового проекта	Рабочий веб-интерфейс, позволяющий сконфигурировать и скачать кастомный хук	Январь 2026
Доработка, оптимизация и финальное тестирование	Реализация дополнительных типов алгоритмов, оптимизация газового потребления контрактов. Подготовка финальной документации	Библиотека с минимум 3 типами хуков. Итоговый отчет с результатами тестирования и документация. Программа готова к релизу	Февраль 2026
Релиз и внедрение	Публикация исходного кода библиотеки в открытом репозитории (GitHub). Развертывание веб-интерфейса	Открытая библиотека смарт-контрактов, доступная для использования разработчиками	Март 2026

2. Используемый технологический стек и его обоснование

2.1 Перечень используемых технологий:

Технология/ Инструмент	Описание	Причины выбора
Ethereum	Блокчейн с поддержкой смарт-контрактов.	Широкое использование, большое число децентрализованных бирж, высокое количество токенов
Solidity	Язык смарт-контрактов	Высокая популярность и множество библиотек для взаимодействия со смарт-контрактом.
Foundry (Forge, Anvil, Cast)	Современный фреймворк для разработки на Solidity	Обеспечивает высокую скорость выполнения, что критично для итеративной разработки и сложных симуляций.
FoundryUp	Утилита для управления версиями Foundry	Обеспечивает удобную установку всех зависимостей фреймворка Foundry
Uniswap V4 Core & Periphery	Базовый фреймворк для создания DEX и хуков	Использование V4 обусловлено его хуковой системой, которая позволяет внедрять пользовательскую логику (динамические комиссии) напрямую в пулы.
Chainlink Price Feeds	Децентрализованные данные о ценах токенов на централизованных биржах	Позволяет смарт-контрактам безопасно получать в реальном времени данные с централизованных бирж.
Binance API	Application Programming Interface для получения исторических данных о ценах на криптовалюты	Позволяет получать доступ к истории изменения цен на токены и на их основе проводить тесты смарт-контрактов
OpenZeppelin Contracts и UniswapHooks	Библиотеки с шаблонами смарт-контрактов	Официальные шаблоны для хуков, ускоряющие разработку и снижающие риски уязвимостей.
Flask	Фреймворк для создания веб-интерфейсов на Python	Простота в разработке, гибкость, возможность расширения функционала

3. Критерии оценивания проекта

Критерий	Описание
Функциональная полнота	Реализация запланированного количества типов хуков динамических комиссий (минимум 3 различных алгоритма)
Функциональность - Процент выполнения функциональных требований	90% (к моменту финального тестирования все ключевые функции смарт-контракта и сопутствующие интеграции реализованы).
Удобство использования	Работоспособность скриптов развертывания и веб-интерфейса, позволяющих пользователю сгенерировать и получить готовый к использованию контракт
Соблюдение сроков и плана - Процент выполнения работы в срок (%)	Не менее 85% работы выполнено в срок

4. Особые пометки

- Данный проект полностью зависит от экосистемы UniswapV4 и без нее не сможет функционировать
- Риски безопасности оракулов: Алгоритмы, зависящие от Chainlink, подвержены рискам, связанным с доступностью и манипуляцией данными оракула, хотя этот риск минимален при использовании децентрализованной сети Chainlink

- Волатильность рынка: Эффективность алгоритмов может значительно варьироваться в зависимости от рыночного режима (Bear Market, Bull Market)